Fig. 1

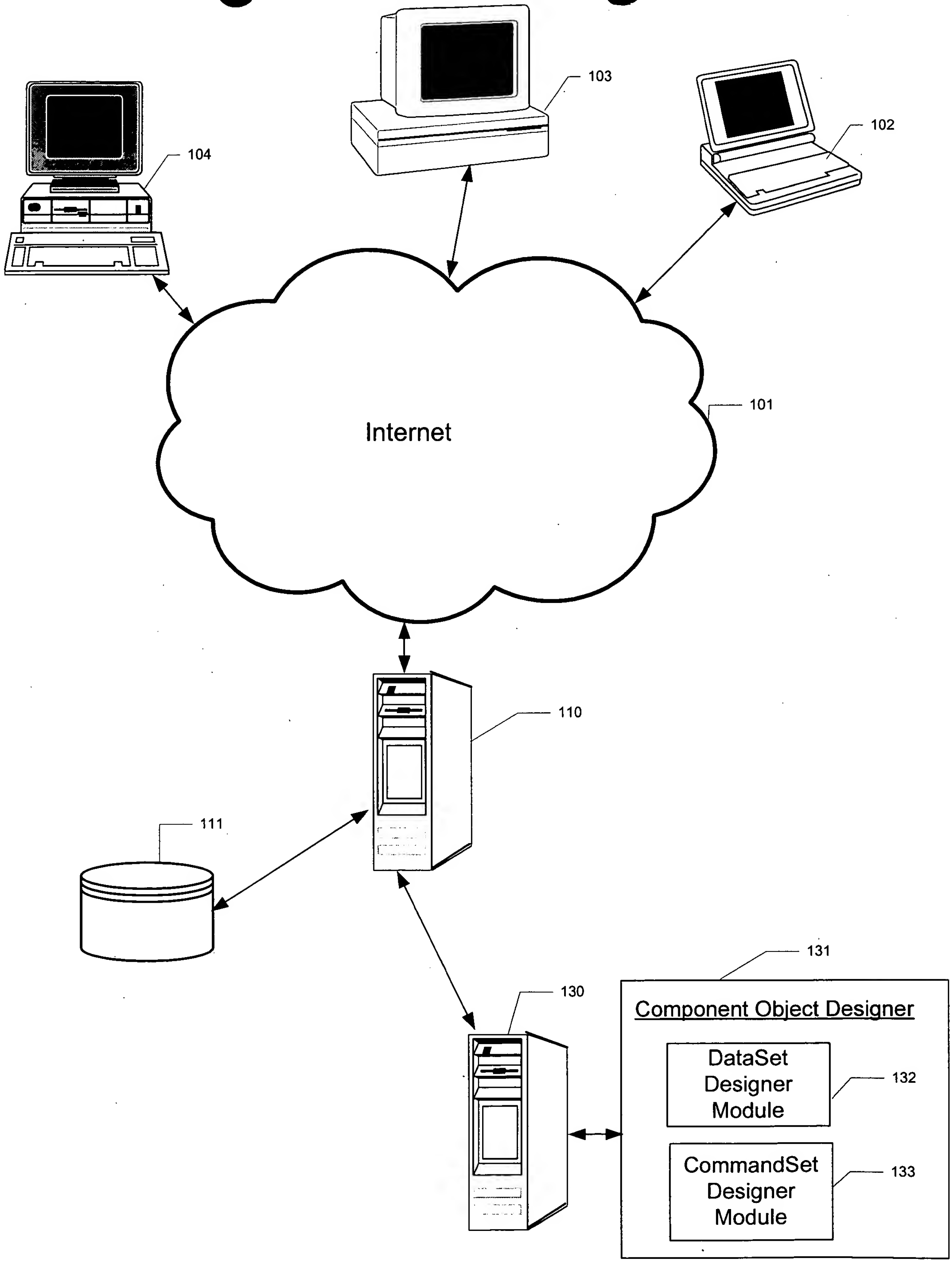
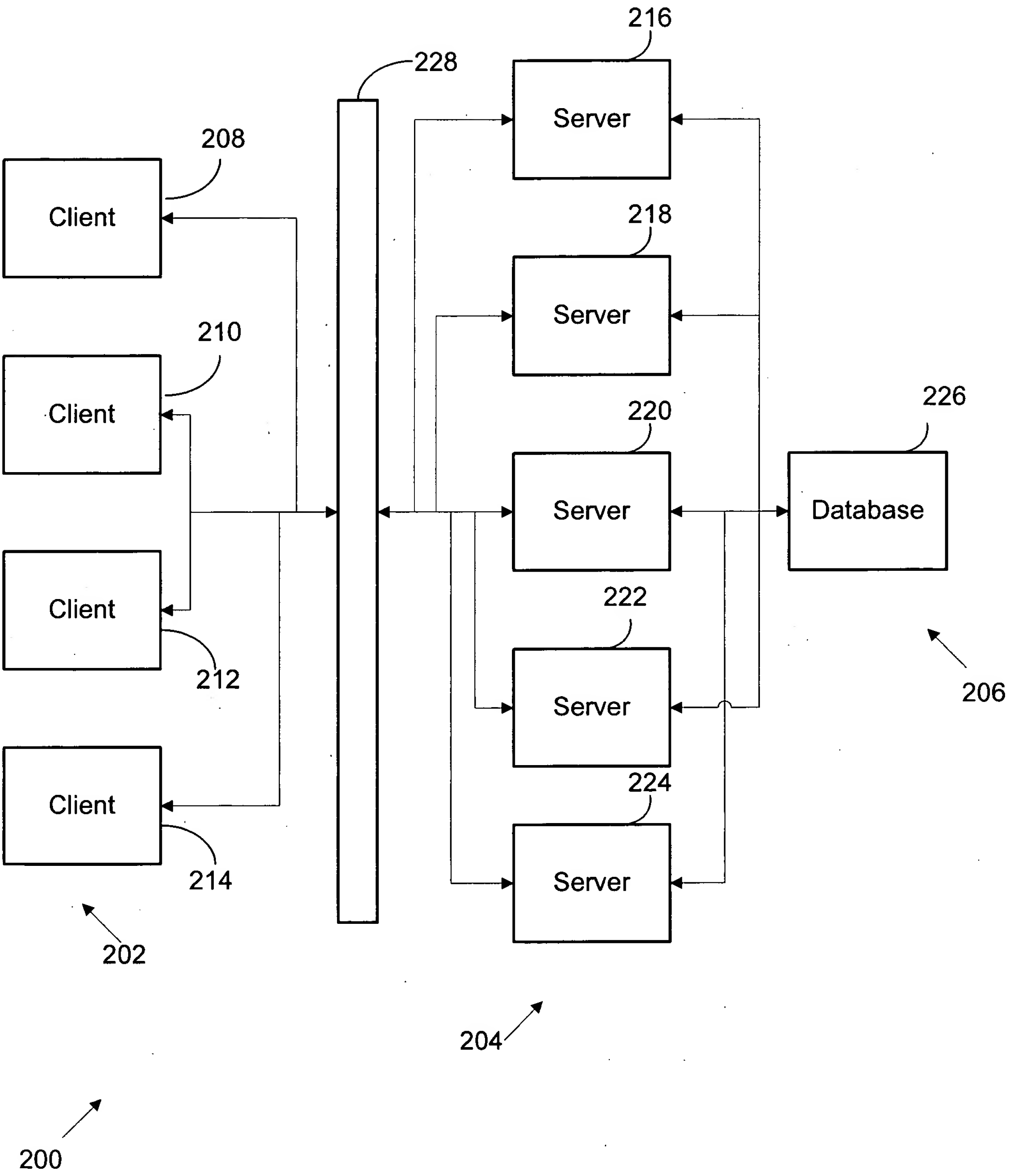

103
102
104
Internet
101
110
111
131
Component Object Designer
130
DataSet Designer Module
132
CommandSet Designer Module
133

Fig. 2



216
228
Server
208
Client
218
Server
210
Client
220
226
Server
Database
222
212
Client
Server
206
224
Client
Server
214
202
204
200

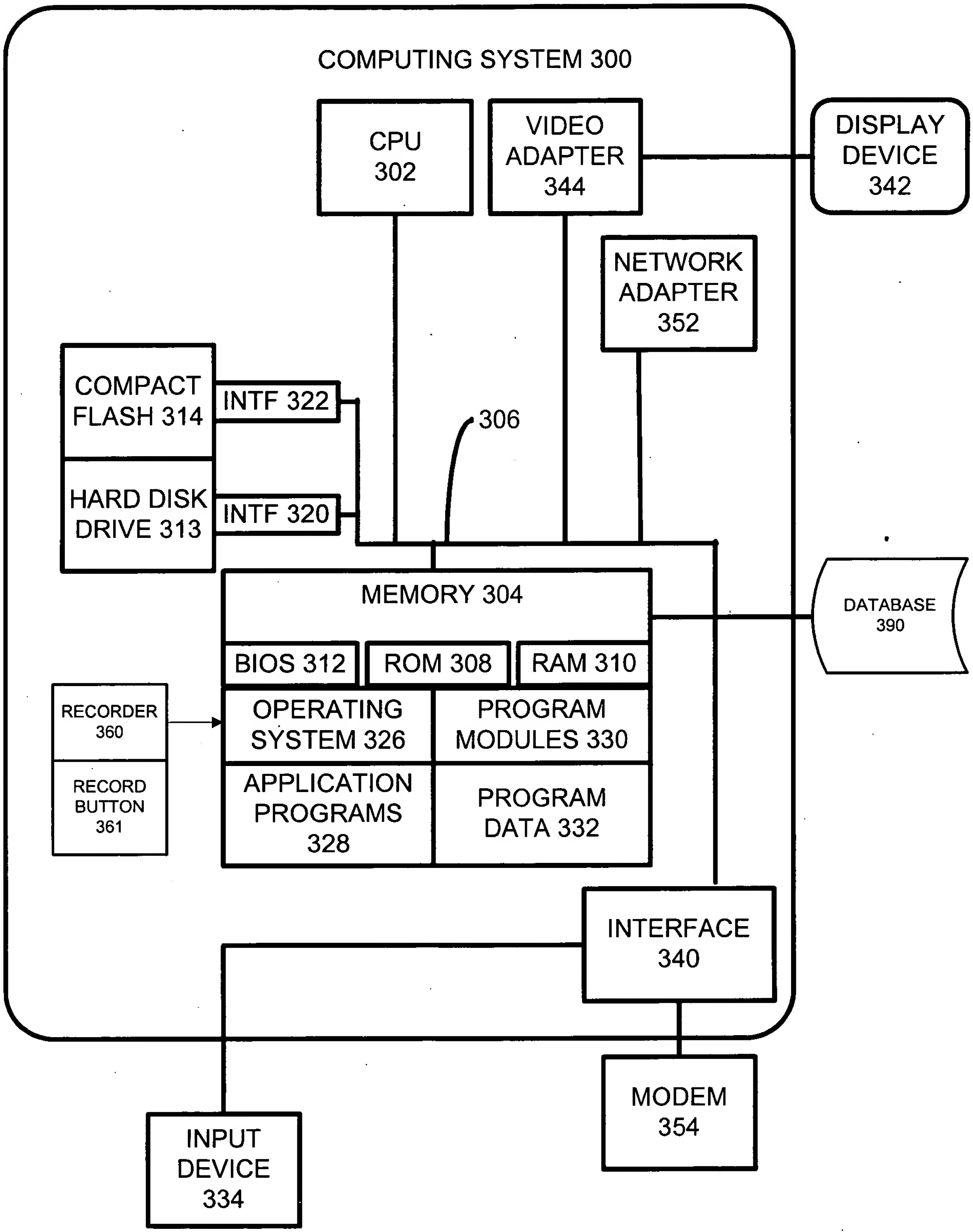
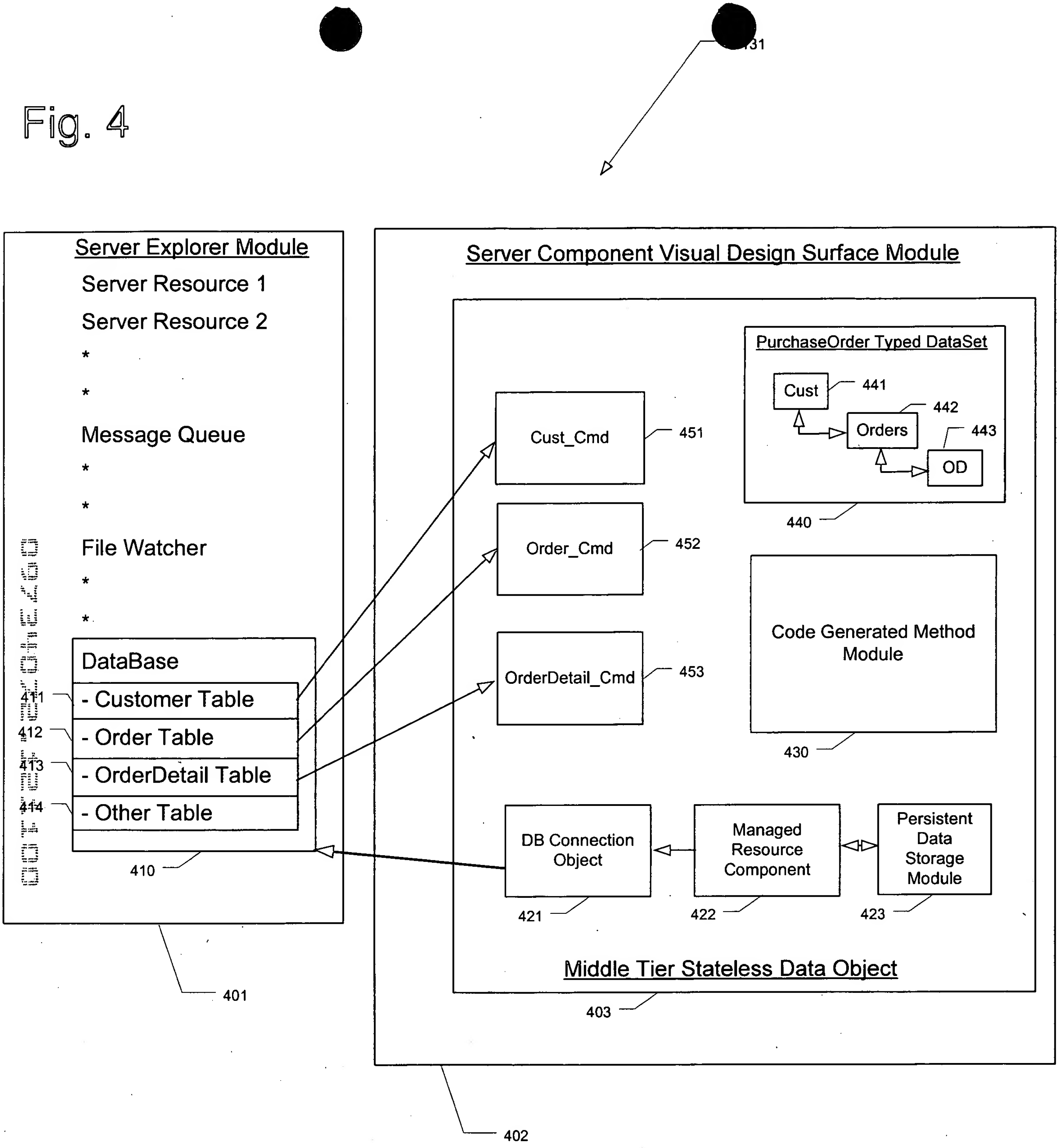

COMPUTING SYSTEM 300
CPU 302
VIDEO ADAPTER 344
DISPLAY DEVICE 342
NETWORK ADAPTER 352
COMPACT FLASH 314
INTF 322
306
HARD DISK DRIVE 313
INTF 320
MEMORY 304
DATABASE 390
BIOS 312
ROM 308
RAM 310
RECORDER 360
RECORD BUTTON 361
OPERATING SYSTEM 326
PROGRAM MODULES 330
APPLICATION PROGRAMS 328
PROGRAM DATA 332
INTERFACE 340
MODEM 354
INPUT DEVICE 334

Fig. 3

Fig. 4



431
Server Explorer Module
Server Resource 1
Server Resource 2
*
*
Message Queue
*
*
File Watcher
*
*
DataBase
- Customer Table
- Order Table
- OrderDetail Table
- Other Table
411
412
413
414
410
401
Server Component Visual Design Surface Module
PurchaseOrder Typed DataSet
Cust
441
Orders
442
OD
443
440
Cust_Cmd
451
Order_Cmd
452
OrderDetail_Cmd
453
Code Generated Method Module
430
DB Connection Object
421
Managed Resource Component
422
Persistent Data Storage Module
423
Middle Tier Stateless Data Object
403
402

Fig. 5

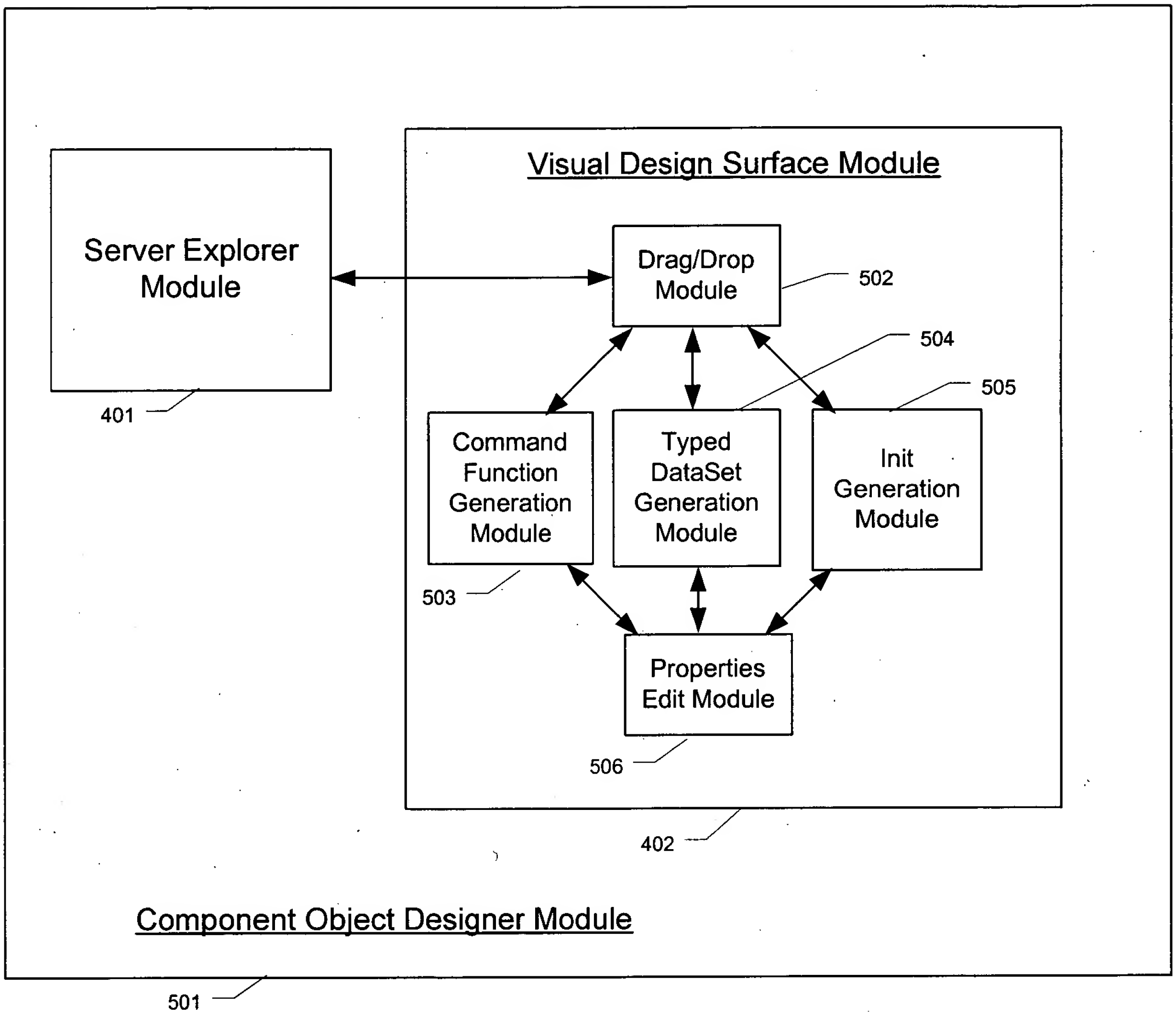

Server Explorer Module
401
Visual Design Surface Module
Drag/Drop Module
502
504
505
Command Function Generation Module
Typed DataSet Generation Module
Init Generation Module
503
Properties Edit Module
506
402
Component Object Designer Module
501

Fig. 6

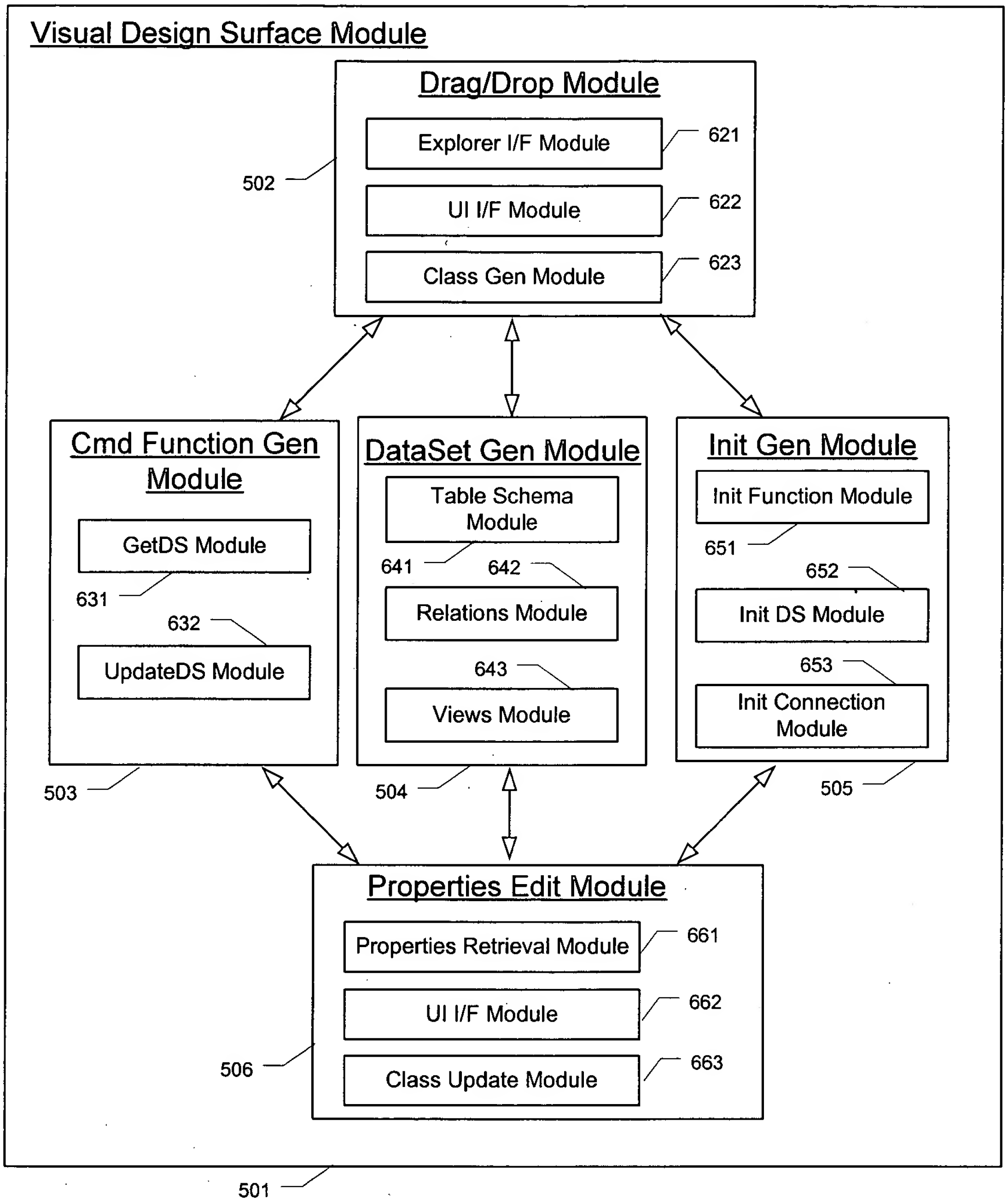

Visual Design Surface Module
Drag/Drop Module
Explorer I/F Module
621
UI I/F Module
622
Class Gen Module
623
502
Cmd Function Gen Module
GetDS Module
631
632
UpdateDS Module
503
DataSet Gen Module
Table Schema Module
641
642
Relations Module
643
Views Module
504
Init Gen Module
Init Function Module
651
652
Init DS Module
653
Init Connection Module
505
Properties Edit Module
Properties Retrieval Module
661
UI I/F Module
662
Class Update Module
663
506
501

Fig. 7a

FillDataSet Method

```
Public Function XXXXFillDataSet(ByVal dataSet As DataSet{,ByVal {ParameterName} As System.{ParameterType}}) As Long

' Copy input Parameter Values to the Parameter Collection
        DIM vtParameter As Varient
        vtParameter1 = {ParameterName}
        Set {Adapterinstance}.SelectCommand.Parameters.Item(ParameterName.Value=vtParameter1
'Create an error handling block in case filldataset throws an exception
        Try
'Load dataSet from the command
        XXXXXFillDataSet = {AdapterInstance}.FillDataSet(dataset)
        Catch eFillExecption as Exception
                'if we can do a debug.print without a reference, then we should print the
                'error info to the output window and re-throw the error.
        End

End Function
```

701

Fig. 7b

UpdateDataSet Method

```
Public Function XXXXUodateDataSet (ByVal updateDataSet As DataSet) As Long

        const DataCursorRowState flag = DataCursorRowState.new | DataCursorRowState.Deleted |
DataCursorRowState.ModifiedCurrent;
        DataRow[] datarows = dataTable.Select(" ", " ", flag);

        Dim UpdateRows as DataRows()
        Dim NewRows as DataRows()
        Dim DeletedRows as DataRows()

' Get All of the Deleted Rows and Update the Data Store
        DeletedRows = UpdateDataSet.Select(" ", " ", DataCursorRowState.Deleted)
        If 0 < dataRowLength then
                Update(DeletedRows, AdapterInstance.DefaultSourceTableName);
        End If

' Get All of the Updates Rows and Update the Data Store
        UpdatedRows = UpdateDataSet.Select(" ", " ", DataCursorRowState.ModifiedCurrent)
        If 0 < dataRowLength then
                Update(DeletedRows, AdapterInstance.DefaultSourceTableName);
        End If

' Get All of the New Rows and Update the Data Store
        NewRows = UpdateDataSet.Select(" ", " ", DataCursorRowState.New)
        If 0 < dataRowLength then
                Update(DeletedRows, AdapterInstance.DefaultSourceTableName);
        End If

End Function
```

702

Fig. 8a

FillAllDataSet

```
Public Sub FillAllDataSet(ByVal dataset As DataSet(, ByVal Parameters As System.{ParameterType})

        ' The line below is added for each adapter on the Virtual Design Surface
        ' Passing into the function only the necessary parameters
        (AdapterName1)FillDataSet dataset{, _parameters}
        (AdapterName2)FillDataSet dataset{, _parameters}
        *
        *
        *
        (AdapterNameN)FillDataSet dataset{, _parameters}

End Sub
```

802

801

Fig. 8b

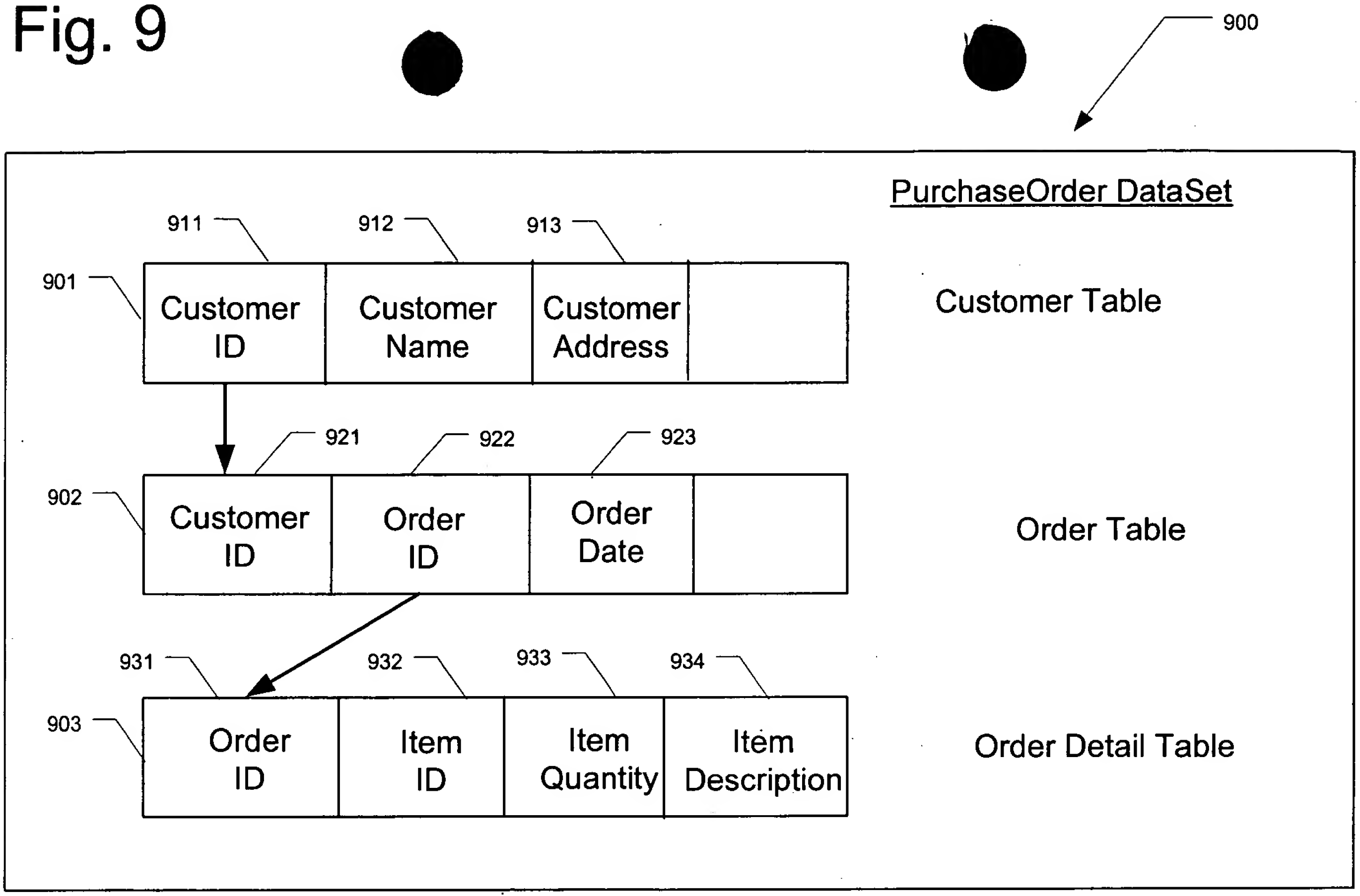
UpdateAllDataSet

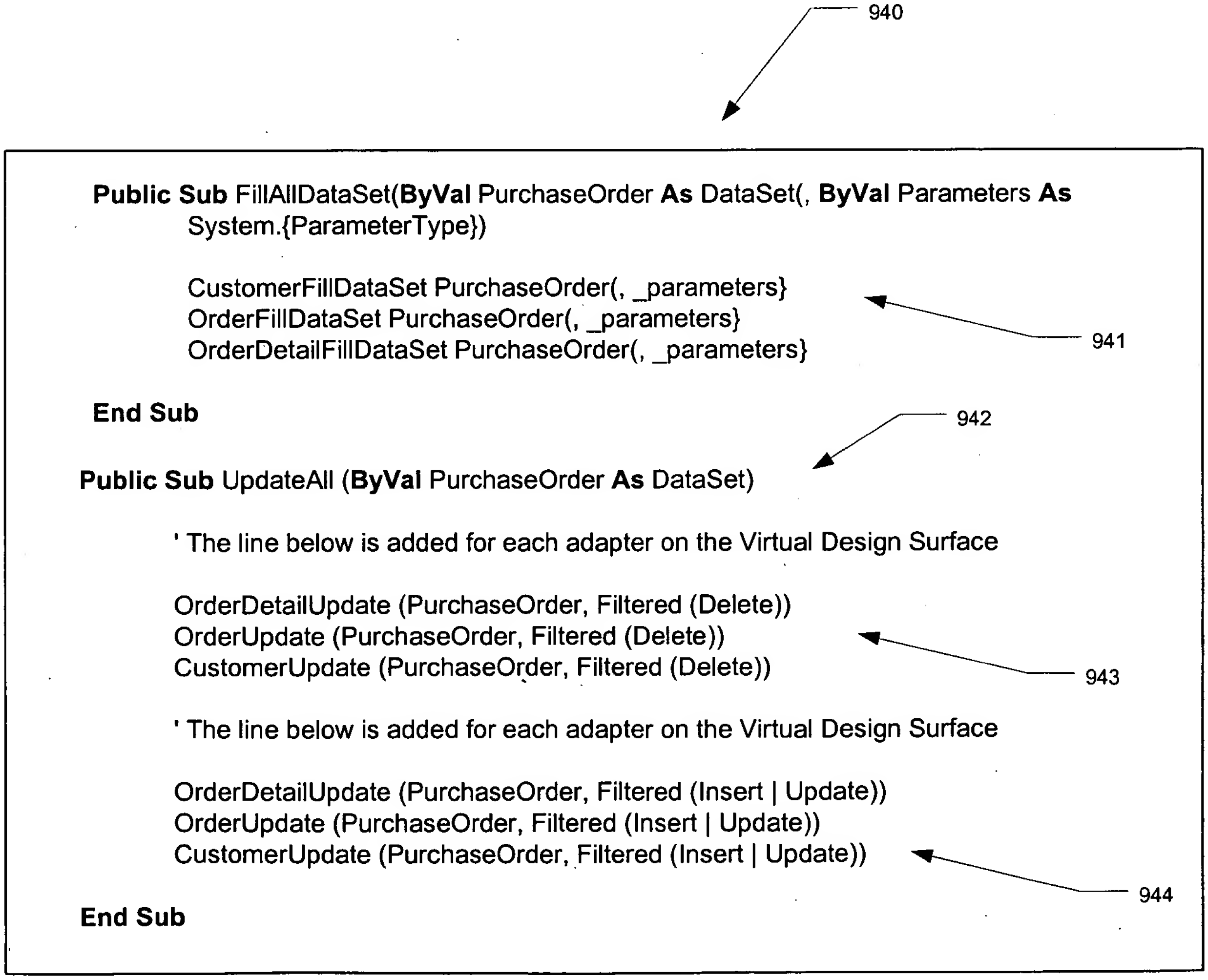
```
Public Sub Update{DataSetName} (ByVal dataset As DataSet)

        ' The line below is added for each adapter on the Virtual Design Surface

        (AdapterName1)Update dataSet
        (AdapterName2)Update dataSet
        *
        *
        *
        (AdapterNameN)Update dataSet

End Sub
```

812

811

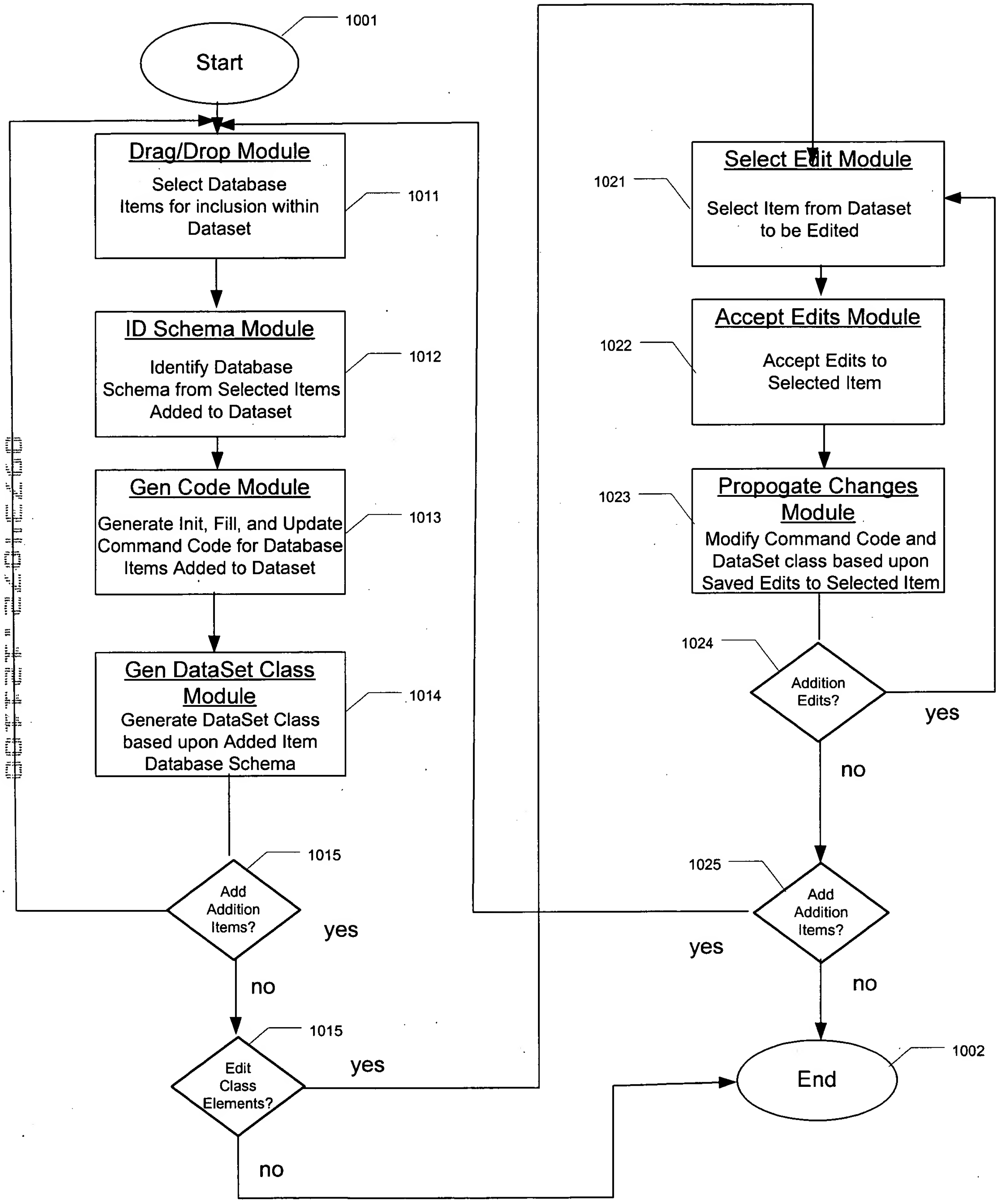
Fig. 9




940
941
942
943
944

```
Public Sub FillAllDataSet(ByVal PurchaseOrder As DataSet(, ByVal Parameters As
        System.{ParameterType})

        CustomerFillDataSet PurchaseOrder(, _parameters}
        OrderFillDataSet PurchaseOrder(, _parameters}
        OrderDetailFillDataSet PurchaseOrder(, _parameters}

End Sub

Public Sub UpdateAll (ByVal PurchaseOrder As DataSet)

        ' The line below is added for each adapter on the Virtual Design Surface

        OrderDetailUpdate (PurchaseOrder, Filtered (Delete))
        OrderUpdate (PurchaseOrder, Filtered (Delete))
        CustomerUpdate (PurchaseOrder, Filtered (Delete))

        ' The line below is added for each adapter on the Virtual Design Surface

        OrderDetailUpdate (PurchaseOrder, Filtered (Insert | Update))
        OrderUpdate (PurchaseOrder, Filtered (Insert | Update))
        CustomerUpdate (PurchaseOrder, Filtered (Insert | Update))

End Sub
```

Fig. 10



1001
Start
Drag/Drop Module
Select Database Items for inclusion within Dataset
1011
ID Schema Module
Identify Database Schema from Selected Items Added to Dataset
1012
Gen Code Module
Generate Init, Fill, and Update Command Code for Database Items Added to Dataset
1013
Gen DataSet Class Module
Generate DataSet Class based upon Added Item Database Schema
1014
1015
Add Addition Items?
yes
no
1015
Edit Class Elements?
yes
no
1021
Select Edit Module
Select Item from Dataset to be Edited
1022
Accept Edits Module
Accept Edits to Selected Item
1023
Propogate Changes Module
Modify Command Code and DataSet class based upon Saved Edits to Selected Item
1024
Addition Edits?
yes
no
1025
Add Addition Items?
yes
no
1002
End